A Distributed Voice over Internet Protocol and Public Switched Telephone Network Honeynet Framework

An Undergraduate Thesis

Presented to the Academic Faculty of

The College of Computing.

By

Zachary Hanif

In partial fulfillment

Of the requirements for the degree

B.S. Computer Science with Research Option

Georgia Institute of Technology

May 2009

Table of Contents

Terms and basic assumptions:

To gather usable information from this paper, certain terms will be used to signify specific ideas. Herein, the word “spammer” is defined as a user of a messaging technology who makes unsolicited, often blind and illicit communications between himself, and/or an automated agent of himself, and a traditional user of this messaging system. The word “attacker” is defined as an individual who seeks to exploit the underlying structure of the aforementioned messaging system for ends that are assumed to be malicious as outlined previously.

The abbreviations VoIP and PSTN refer to Voice over Internet Protocol Technology and Public Switched Telephone Network, respectively. These terms define two different networking protocols which are able to interface with each other through specialized hardware and software. These protocols are commonly defined by the software and hardware that are often distinct and difficult to entirely integrate.

The abbreviation PBX refers to a Public Branch Exchange, a device that routes internal calls within a privately owned telephone network. These exchanges are often used in businesses to route internal calls without having to directly utilize the external publically accessible telephone network.

While the terms attacker and spammer are not mutually inclusive or exclusive by definition, they can often be one and the same. While these terms can mean different things, for the purposes of this paper, they both have negative connotations. While the annoyance of bulk messages is immediately apparent, it is critical to note that such messages can tax a myriad of computing resources, from storage space, to bandwidth limitations. Furthermore, the dangers to individual users exist in the form of outright deceptions designed to rob the victim of their personal information or monetary wealth.

It is important to remember that VoIP is not one discrete protocol or service, but represents a collection of protocols and services that are both open sourced and proprietary. Often, individuals or groups of

individuals create their own implementation of the current spec to suit their individual needs. These implementations include H.323 protocols, along with Skype and Vonage's respective implementations. Likewise, PSTN connections are not handled through one singular entity, but are a collection of standards that have been agreed upon by national and international telephone service providers.

Abstract:

Because of the recent advent of Voice over Internet Protocol technologies, security researchers have had difficulty keeping up with the rapid increase in such electronic spam and related security threats. As a response, Lightning Dog was conceived to provide researchers with the tools and framework needed to gather immediate data from the wild, produced by the spammers themselves. As spammers have repeatedly demonstrated that they will attempt to utilize all possible mediums of communication to contact others, the advent of VoIP has provided them with numerous new avenues of propagating their unwanted messages to legitimate users of the communication system.

Lightning Dog is based on a three part system which incorporates baited websites, multiple gathering nodes and a centralized collection and analysis machine for dissecting and cataloging the captured telephonic spam and attack data. This project is particularly notable because it allows researchers to gather and analyze both Voice over Internet Protocol and Public Switched Telephone Network captured telephony spam, thereby allowing researchers to correlate any links between the two mediums of spam publication. As researchers are having difficulty quickly capturing data that accurately reflects either of these two areas of telephonic spam, it is the goal of this project to provide an accessible method to resolve that issue.

Lightning Dog proposes a honeynet dedicated to capturing and recording VoIP and PSTN specific data, allowing researchers easy access to recent attacks that are constructed by telephonic attackers, as opposed to having to rely on out-of-date, third-party, or unreliable information.

In this paper, the system is introduced, its specific motivations are explained, the specific goals and the final design of the system are presented. Finally, proposed usage and deployment scenarios are

presented, along with a review of published related literature and works, and a discussion of the implications and possible future work for this system.

1. **Introduction**:

This section will introduce the project's initial motivation while offering a brief explanation of the need for this project. Primarily, the rapid growth of the VoIP field alongside the increasing legislation to prevent traditional PSTN spam has resulted in a migration of spammers to communication systems that use this new protocol.

VoIP technologies have enjoyed a recent surge in popularity, despite, or perhaps, because of their relative newness the multiple financial incentives, and because of the added convenience this technology brings to interpersonal communication. The rapid advances of these technologies, however, have resulted in exceedingly rapid development of their underlying architecture, attracting malicious entities to utilize VoIP technologies to their own ends.

First conceived in 1974, VoIP protocols were primarily ignored by the general populace and by all but the original developers due to network limitations that hindered transmission speed and efficiency. Beginning in the middle to late 90's further advances in internet protocols led to renewed interest in a system that could begin to inexpensively replace traditional PSTN connections for personal communication. While growth continued from then on, the true adoption of VoIP technologies is more commonly ascribed to 2004, when a number of commercial VoIP companies became well publicized and began offering reliable services in direct competition with more traditional telephonic service providers (Callahan).

There are some specific challenges that prevent more rapid adoption of these protocols and technology. As previously mentioned, the original set of standards went largely unused for decades, as the networks they were designed for were incapable of carrying the level of traffic that such services demanded in order to have a desirable quality of service. Today, VoIP services are somewhat more reliable than they

have been in the past, and are enjoying a surge of popularity amongst communities of young people and large corporations due to their relatively inexpensive cost to deploy and utilize (Callahan). This market is rapidly expanding, and is poised to grow continually for the next several years (Callahan). Due to this surge of interest, there has been rapid advancement in an attempt to compensate for the concerns over the quality of VoIP services; these improvements, in turn, attract more and more people to the service.

PSTNs have existed since the dawn of the telephonic age, and until relatively recently, remained reasonably close to their original implementation. Originally these systems were entirely analogue, relying on manual connections, often existing as purely local exchanges, controlled by a number of private companies. As time passed, the corporations merged and began to anticipate that digital signals would lead to more efficient call routing and quality. Today, with the advent of VoIP technologies, PSTNs have been adapted to allow users to make calls originating from IP based systems to PSTN based systems (Arnold). While the benefits of VoIP adoption are clear to large corporations and are becoming more apparent to the average consumer, attackers are beginning to notice the openings that this new form of communication provides.

Spammers and malicious users of VoIP technologies are primarily attracted to the relative ease of use, inexpensive operating and construction costs, and the relatively unregulated area that these new technologies operate in. The fact that many VoIP services exist which allow for unlimited calling, combined with the reality that it is relatively inexpensive for a dedicated spammer to create such services on his own, have led these attackers to take an interest in the VoIP protocol, specifically in the way it interfaces with currently existing PSTN's. In addition to the economic incentives that spammers are attracted to, the difficulty involved in geographically locating a user during or after a call enables attackers to circumvent legal jurisdiction when placing unsolicited calls to locations that attempt to legally discourage such communications.

The relative youth of the field of VoIP security prevents researchers from monitoring and adapting to the increasing threat that these spammers represent. Due to this problem there are few safety measures in place to protect legitimate users who will often implicitly trust the system to keep them safe because of the similarity to traditional PSTN systems. Because of the increased margin of profit and the increased anonymity that a transfer of operations from traditional phone advertisement to VoIP technologies represents, it has proven to be a natural safe-haven for the migration of these individuals.

Specific Motivation and Intended Research Targets:

> Herein the specific motivation for creating a VoIP honeynet is discussed, along with a discussion on whom this research is intended to affect directly and indirectly. In short, this particular framework structure deemed necessary due to the relative dearth of currently available research data, and as such, provides the greatest direct benefit to telephonic security researchers.

Project Lightning Dog was created to assist researchers in gaining illumination to multiple kinds of telephonic threats and nuisances, and to allow them to quickly and easily construct VoIP and PSTN networks within a laboratory setting, thereby allowing them to compare their theoretical models of proper network construction, architecture, and current security tools and practices against what is currently in place in the business world.

Due to the lack of raw data easily available to researchers who pursue the analysis of VoIP technologies, there is a large gulf of knowledge between security professionals and the systems that they are supposed to secure and monitor. While this is in part due to the relative rarity of VoIP systems in the United States, the problem could be alleviated by creating a Linux distribution with detailed instructions so as to quickly create a honeynet for research data-gathering, along with a set of formatted guidelines

to lead beginning research teams to the specific manner of gathering relevant data in the world of VoIP research (Arnold).

Lightning Dog proposes to be a system that captures and analyzes all telephonic spam that it can attract, including PSTN channeled traffic. For security researchers who are interested in monitoring the activities and techniques that attackers use to exploit protected networks, honeynets have proven to be a reliable, easily monitored, and effective source of information for gathering an attacker's tools, patterns, and signature as it occurs, as well as storing the state of these attacks for future storage, analysis and study. As such, it is this underlying structure that forms the basis for this research framework.

While there is significant interest in VoIP technologies, there is little research being done in the world of VoIP honeynets or intrusion and attack detection systems, and there is little solid data that researchers can utilize and collect at will with which to perform more directed research and draw larger conclusions. At this time, individual researchers are forced to develop their own methods of capturing recorded telephonic information. This fact means that beginning research teams face a hurdle to enter the field, and even veteran research teams run the risk of having to expend significant amounts of energy to keep their methods of data collection and storage running as project requirements change and as their systems evolve (Niccolini 1-9). As a direct result of the lack of useful data that researchers can reliably gather for themselves, researchers have difficulty determining conclusive answers to even the most basic of questions surrounding this field; many are unable to determine just what level of an issue VoIP attacks truly pose, the frequency of these attacks, or the level of specific direction that they use to target individual users.

There is little immediate incentive for the average VoIP end-user within this project. This research more directly benefits the academic world, and through that, there will be more secure technologies created,

which, in turn, will result in a more satisfying, secure, and enjoyable experience for the end-user. It is important to remember that the end-user in this scenario is not just the user of services such as Skype or more open VoIP technologies, but indeed, exclusively PSTN users are affected by VoIP technologies to an increasing degree. Ultimately, this project will result in the ability to record and profile attackers, eventually leading to a reduction in the sheer amount of spam calls and limiting the effects of phishing attacks.

Researchers are confronted with the demand to provide means to secure these new technologies while they struggle with a lack of standardization in VoIP deployment, servers, and clients. The intermingling of VoIP and PSTN technologies simply serves to further complicate the matter, as each system has its own security concerns, concerns that can be exacerbated by merging the two systems improperly. To allow security researchers to effectively stay ahead of the ever quickening curve of attacks and vulnerabilities, a collective system must be created to record and analyze these attacks. This system must be robust, flexible, and easily extended to the needs of the researchers. To this end, Lightning Dog was conceived as a distributed, flexible and robust system to serve as a point of attack for spammers and malicious attackers alike.

Problem Description:

This section will detail the design goals and challenges of this specific system along with the particular complexities of the architecture. These goals primarily center on the need to remain both secure and anonymous while being able to attract and capture telephonic spam and attack data and while being adaptable enough to be modified easily by the researchers who are utilizing the system.

Creating any honeynet is challenging due to several underlying reasons; security and the ability to remain unidentifiable as a research tool rank amongst the most important concerns. Further, this particular form of honeynet attracts interesting difficulties that are not addressed in standardized honeynet architectures. Effectively, such a system must, as a subset of a honeynet application, address the inherent problems that are faced by traditional honeynets, while gracefully dealing with the additional problems that this particular architecture creates.

While creating a static honeynet would be interesting, there is already a great deal of work that has been done on the area of single instances of honeynet creation and research. The difficulty in this revolves around supporting a variety of hardware, and allowing the system to be swiftly deployed in an environment that the designers of the system cannot control or depend upon. Additionally, this system will need to support a variety of specialty PSTN hardware which is often outdated, poorly supported, or difficult to configure. This has led to a number of specialty requirements for this system. Despite the difficulties in constructing such a honeynet, particularly one that captures information from such non-traditional sources as PSTNs, it is the hope of this project that it would facilitate the capturing of a wider variety of telephonic spam then other, more orthodox systems.

Most importantly this system, just as any honeynet, needs to be able to withstand the repeated attacks that it attempts to attract and record. The compromise of a system could potentially lead to discovery that the entire node, or worse, the entire network, is a trap designed to record an attacker's behavior. Furthermore, the potential that an attacker could manage to corrupt or view the currently recorded attack data is a threat that cannot be discounted. Additionally, that honeynet must also be prepared to fool an attacker into believing that it is not a monitoring system, but is instead an entirely legitimate system that an attacker would be interested in interacting with in a natural way. For standard honeynets, this typically means that the system must respond to direct attacks upon its hosted services.

A major part of this research revolves around determining what exactly constitutes “standard behavior” for an enterprise VoIP system. This project plans to alleviate this issue for researchers by handing them the foundational building blocks of an enterprise VoIP network. By allowing the researchers to quickly construct and deconstruct models of real-world VoIP networks alongside their theoretical designs and models, they will be able to more swiftly and flexibly create scenarios and test-beds for their research, as well as gather information in this realistically simulated environment.

In addition to the technical difficulties inherent within creating honeynets, this particular variant presents a difficult issue. Whereas virtually every internet-facing computer daily confronts hundreds of automated attempts to illicitly access traditional computer systems, VoIP spam needs a more focused audience. Currently, it is uncertain exactly how these attackers select their targets, and it is one of the goals of this project to determine exactly how targets are selected. Lightning Dog intends to resolve this difficulty through the use of specifically constructed bait-sites, deployed along with the proliferation of numerous, unique, and recorded SIP URI’s throughout the increasingly myriad social networking websites that are often harvested for traditional spamming purposes.

The specific remedies to these solutions that the Lightning Dog team has integrated into the systems architecture are outlined in the next section.

Design and Architecture Principals:

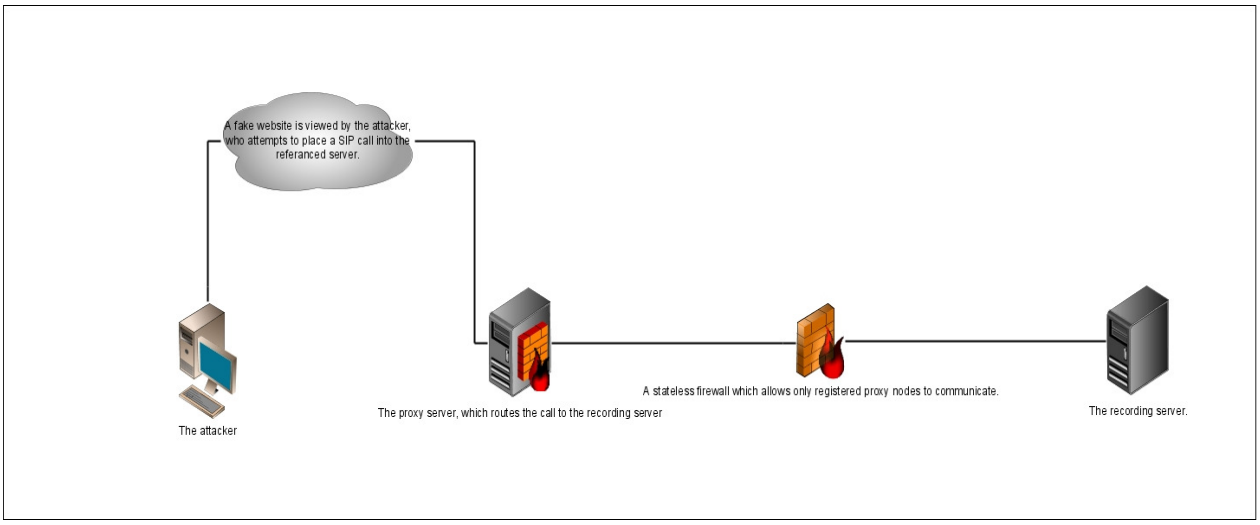
> This section introduces the elements of the overall system, breaking them down into discrete parts. These parts are best summarized as the bait websites, the proxy nodes, and the analysis and collection server. As is apparent in the following section, the system is highly modular due to what compromises the component parts.

There are two major venues of utilizing VoIP calls to propagate spam and fishing attacks that are relevant to the greater research community, both of which this project intends to capture and analyze. Firstly, attackers are utilizing VoIP technologies to make computer to computer calls, their traffic entirely transmitted over TCP/IP networks (Abdelnur). The threats that this sort of spam presents are multifold, subtle, poorly understood. Secondly, attackers are using VoIP systems as the originating point to transmit messages to users of traditional PSTNs while managing to avoid legislation designed to hinder these transmissions.

Lightning Dog is designed to operate in a distributed environment, and has been designed to scale easily and efficiently; a concern that is raised to a heightened degree due to the nature and bandwidth demands of the particular type of traffic that this system proposes to capture (Fiedler 11-17). To this end, the system has been designed into three major parts, thereby creating three separate "modules" that can be deployed in various means. These parts are the faked websites, the proxy nodal systems, and the analysis and collection server. This system has been chosen because it is robust, easily modified, and simple to deploy. Lightning Dog is a two-stage deployment, and it focuses on two types of telephonic spam, that which passes over VoIP and that which passes over PSTNs. This first section will detail the architecture that supports VoIP collection and analysis.

The default and most common deployment of this framework is likely to be as displayed in the figure below. Herein, we can see an example network comprised of the attacker, the proxy server, and an expanded view of the recording server's firewall before connecting with the server itself. The images within the diagram represent the following concepts: the attacker is represented by the workstation icons, the larger internet is represented by grey clouds, the proxy nodes are represented as servers with a brick wall and flame attached to the sides, the expanded view of the analysis server's firewall is

represented by the lone brick wall with a flame, and the analysis server itself is represented by the plain black server icon.



Perhaps most important to the system are the individual collection nodes that Lightning Dog uses to gather VoIP spam and attack data. The nodes themselves are little more than ad-hoc computers or virtual machines which merely require a recent version of the Perl programming language and runtime environment installed. Upon these computers, a forwarding and proxying script is run to route traffic that is directed towards these nodes towards the central analysis server. This proxy only transmits packets from a particular port, and it only transmits packets that correspond to common SIP and RTP ports. The proxy supports all the major types of VoIP communication, including video connections, audio transmissions, and it includes functionality that allows a user to register accounts on the analysis server, should that be deemed necessary. Finally, the proxy itself has functionality to record and transmit copies of the packets it transmits, along with a statistical breakdown of other relevant information.

This system has been designed to be deployed on any given resource that the user has at hand, even if that system is not directly controlled by the researcher or his team. This allows for a highly distributed system, surpassing geographic and political boundaries, and allows researchers to more effectively

appear to be a legitimate entity to spammers. As a result, concern about the location of logged information was taken into consideration. In the event that the proxy server is deemed not secure enough to hold the logs that the proxying script generates, it can transmit these logs back to the central analysis server for more secure storage.

Achievement of anonyminity is relatively simple for this portion of the system. Due to the manner in which our VoIP proxy handles incoming connections and manipulates packets, we have effectively created a system that, to any normal user, appears to be the true endpoint of their call. An attacker could preform two types of transmission analysis, should he become suspicious about the host he is interacting with; these analyses were specifically considered during the design of this section of the overall Lightning Dog system. Firstly, it was assumed that the spammer could attempt to analyze the packets themselves that are transmitted from the proxy on return from the analysis server to the attacker.

During the design, there was the concern that, should the proxy system be hosted on a different operating system than the one that hosts the analysis server, tools like p0f could detect the subterfuge. In an effort to combat this, it was decided that traditional packet forwarding techniques were insufficient. The proxy, therefore, does not simply forward the packets it receives, but instead it entirely recreates them, therefore making them appear to have been created by the proxy server itself. In the worst case scenario, an attacker would only be able to determine that the system that he is communicating with is behind a NAT; something not uncommon, and unlikely to arouse undue interest. Additionally, unlike other SIP proxies that are available, this proxying system was created to forward both the SIP and RTP packets that it receives during the initial session negotiation phase. Other proxying systems often only handle the SIP transmissions that occur at the beginning of the session, allowing the

PBX server and the initiator of the call to form a direct connection to each other when transmitting the RTP information. For obvious reasons, this was deemed to be inappropriate behavior for this system.

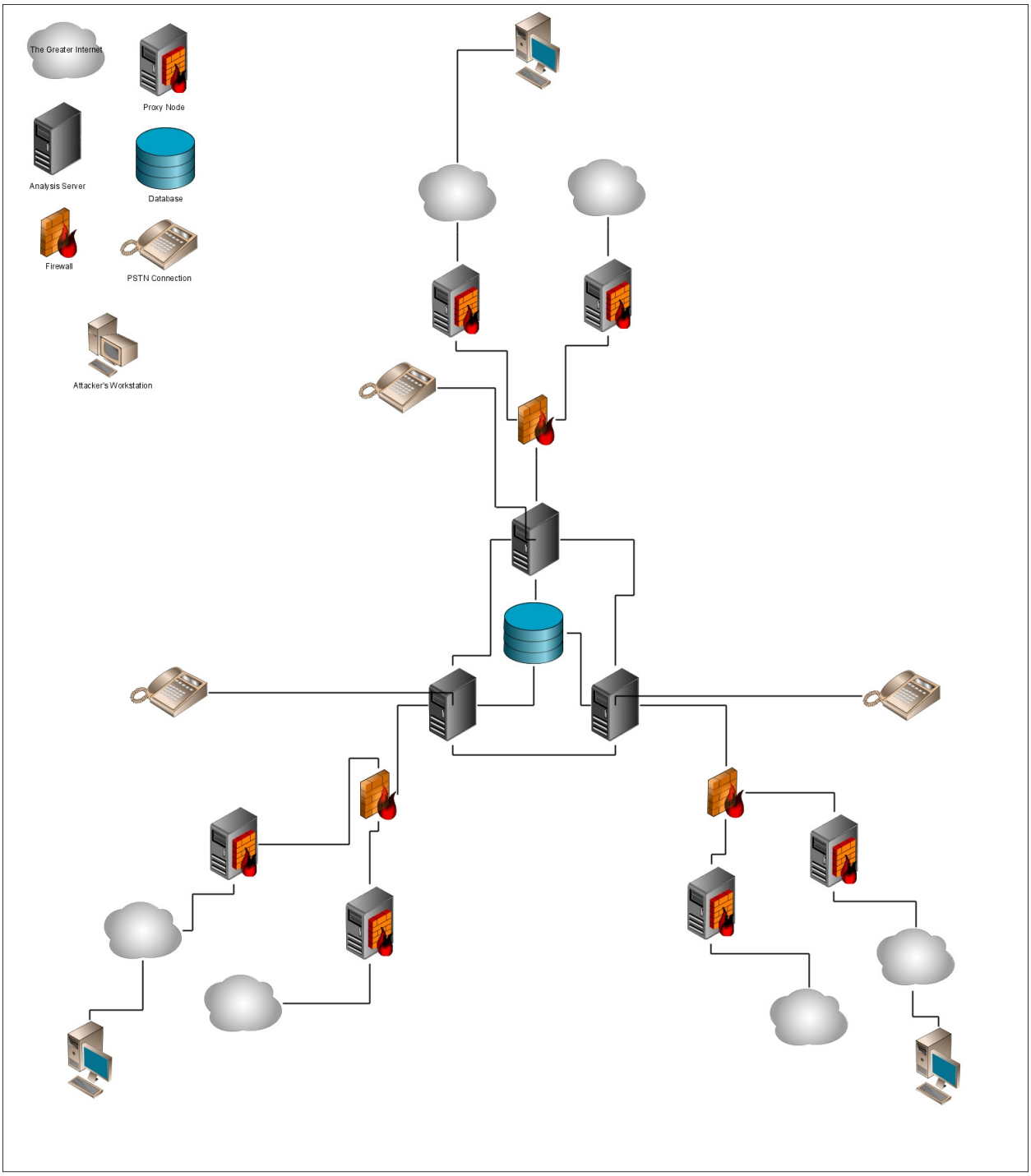
Should a spammer attempt to actively scrutinize the proxy host via port scanning or other techniques, he will simply find a machine that appears to host a normal SIP or RDP proxy, as would be logically be used to forward VoIP calls through an internal network. The simplicity of the forwarding script works to prevent exploitation from the vector of the forwarding node, and combined with the restrictive set of IPTABLES rules that can be deployed on both the forwarding node and the server itself, should provide a resilient means of protection against malicious attack against the system itself. Communications to the server through PSTN messages are easily controlled based on the particular responses that the internal PBX system returns to the caller.

Secondly are the bait-sites, preconfigured websites posted under a generic hosting provider that supports custom SRV record creation in order to attract spammers to what they believe is a business that is legitimately hosting a VoIP service for its internal or external contact information. These sites are designed with multiple layers of information so as to determine exactly how spammers are gathering VoIP information. These methods are designed after the major phases of spam address collection that have been observed in traditional email spam operations. Firstly, the websites display a particular SIP URI in clear, obvious text. This URI corresponds to a discrete voicemail inbox on the collection system, and all calls made to it come exclusively from that URI. Calls to this number from a spammer can be easily deduced as having been gathered with the human eye, or a basic web-scraper. It should be noted that this is the only mailbox that should logically have a chance of collecting any data from a duped member of the public who is legitimately attempting to use the purported service. The second most common method of crawl-identification exists in the form of a SIP URI posted in text which is an identical color to that of the web page's background. This technique, often called white-on-white text,

would provide a strong suggestion that a potential spammer is using a harvesting script to extract URI's from target websites. Other techniques that are designed to capture such scripted behavior include placing SIP URI's into the comments of the webpage in plaintext, hashed formats, and placing such information as the contact info upon images, alt-text, and other various places that would be useful for determining how exactly spammers data-mine this information. These bait-sites, aside from passively recording information on the methods that spammers use to collect information, provide researchers a degree of freedom in which to act in an anonymous fashion; allowing them to directly choose the avenues that they attempt to penetrate, allowing them to determine for themselves exactly how much effort they desire to invest into the charade. The logical gamut runs from the minimal of a handful of sparse sites to the elaborate, such as an entirely fabricated online community of VoIP enthusiasts.

Lastly, the processing and collection server exists to retain and analyze the final pieces of information that the gathering nodes have sent to it. This system is a bare-bones machine running a customized version of the Linux operating system. While the system is designed to be anonymous due to the use of the gathering nodes, the system has been designed to specifically block all incoming and outgoing traffic to any computer that is not one of the configured gathering nodes. A simple set of IPTABLES rules are sufficient to accomplish this task. Furthermore, the system has been stripped of as many superfluous programs as possible, with the design goal that the system should be exceptionally lightweight, and therefore, difficult to attack. The system itself possesses little more than a large hard drive, a processor of moderate speed, a standard 100mb network card, and a Digium TDM400P voice linmodem. Here, the data is collected and recorded for later analysis. Due to the underlying technologies in these servers, they can, if needed, be networked together to share traffic between them, allowing for both traffic and storage load-balancing should such technologies be deemed necessary by the researchers. In an effort to allow for further extensibility, all data analysis tools have been written to interface with standard

database servers, and the underlying operating system of these collection servers allows connections to SANs for extra storage space, should it be needed. The figure below depicts a fairly complex implementation of the entire system, incorporating PSTN connections, VoIP transmissions, and the ability of the servers to collectively record their findings to a central database. The images within the diagram represent the following concepts: the attacker is represented by the workstation icons, the larger internet is represented by grey clouds, the proxy nodes are represented as servers with a brick wall and flame attached to the sides, the expanded view of the analysis server's firewall is represented by the lone brick wall with a flame, and the analysis server itself is represented by the plain black server icon. Further icons, the phone and blue stacked circles, represent a PSTN based attacker and a central database server, respectively.



While the hardware of the analysis server is important, more significant is the ability of the system to handle a large call load, as well as the ability of the currently available tools to process a non-trivial amount of recorded data. Testing has been done to establish a logical upper bound of simultaneous

connections that the system can withstand, along with an analysis of the efficiency of the inspection and reporting tools that are used.

Secondly are the aspects of the system that exist to support capture and analysis of PSTN spam. Currently available kernel modules and Zaptel drivers for Asterisk PBX software were deemed efficient and secure enough to be used for this project, and were used in the final production of the system. Thanks to service providers modern Enum servers, along with the internal procession of the Asterisk server, incoming calls from PSTN numbers are transparent outside their original configuration of the system, allowing researchers to simply utilize the recordings of the collected calls without the concern of having to convert these recordings between multiple formats before gaining the ability to utilize the information.

As a matter of operational security, many telephone service providers allow the forwarding of standardized telephone numbers across geographic boundaries. As a result, we can allow the PSTN calls to terminate directly at the analysis machine, assured that such calls will not be correlated with any particular honeynet effort. Further obfuscation can be achieved by contracting forwarding numbers with local telephone service providers in the regions where the imposter businesses are purported to be located, thereby adding another layer of security to the system by forcing it to route multiple times through the providers' internal network.

Ultimately, it is the modular structure of this network, along with the fact that PSTN networking capabilities are integrated into the system that makes this framework significant. This aspect of the system allows it to function properly as a collected system, as opposed to simply a set of individual nodes whose data requires manual correlation. Here, the design goals revolved around creating a system that was familiar to researchers, efficient, and easily modified to meet the demands of

frequently harried researchers. In an effort to ensure that this system is agile, the base of the systems is all open source, allowing for researchers to still extend the platform to suit their needs, and giving them a large, pre-created community for them to interact with, in the event they need to alter the system in an unfamiliar way in a minimal amount of time. Furthermore, the system has been tailored to utilize some of the most common elements of VoIP and PSTN technology. The selection of Asterisk PBX was founded primarily on the fact that it has a large, well-established community built around it, thereby ensuring that the code is maintained by multiple sources, and providing a large network of other users for researchers to utilize should it be necessary. Secondly, the system has built-in support for the majority of known voice modems, allowing for a minimal amount of initial configuration, and decreasing the need to install a multitude of programs and custom kernels to simply force the system to adapt to the current project at hand. Additionally, the systems are easy to update and maintain, thereby keeping the amount of active maintenance to a minimum.

Anticipated Usage and Deployment Scenario:

> This section explains simple examples and event flow that occur when an attacker begins to interact with the system. Furthermore, it details events that occur when a researcher receives the system and desires to begin the work of deploying it upon his network.

This proposed system is intended to record simple messages, as previously outlined, however, it is entirely capable of detecting and recording more direct attacks such as denial of service attacks, attempts at brute-forcing the passwords of the PBX system to gain access to the repository of hosted voicemail recordings, and the acceptance of vishing attacks as well. Furthermore, the system is designed to passively record the means by which an attacker gathers the contact information of his victims,

allowing researchers to design methods for instructing users on how to protect their contact information from automated harvesting scripts.

To begin with, a researcher wishing to deploy the system will first begin by constructing a relatively simple analysis server. In the prototype constructed as a part of this research, the server was constructed using entirely consumer grade parts; the most unusual aspect of the system was a 4-port linmodem. He would then configure the installed Asterisk server to ensure that the recorded files are saved in an applicable format, and to ensure that the reporting tools have a database or other central repository to store their logs. Secondly, the researcher would find a location to host an instance of the forwarding proxy that is not associated with himself or his community. In the prototype, a virtual machine was rented from Slicehost.com for this purpose. This researcher would then deploy the proxy, and after the initial configuration, edit the analysis servers IPTABLES configuration to allow communication. Lastly, the bait-site would be created by the researcher or his team. In the prototype, the webhost and DNS provider was godaddy.com, chosen for their size, and their ease of altering the default SRV records that they publish. After the system is deployed, it is anticipated that a spammer will crawl the site, believing it to be a standard webpage.

General interactions with the honeynet will likely involve a spammer attempting to locate numbers from one of the bait sites. This spammer has multiple avenues available to him when browsing one of the configured bait-sites. Assuming, for the purposes of this example, that the attacker is reasonably sophisticated, he will have a harvesting script that will parse a particular webpage, then automatically establish a call with that number and transmit a recording of the message that the spammer wishes to communicate. This harvesting script would likely be able to derive multiple numbers from the hypothetical bait-site, each one unique, and stored in various manners that correspond to the way they were harvested. For example, we can assume that 123@LightningDog.com is the URI provided in clear

text, and 345@LightningDog.com is the URI that is obfuscated through white-on-white text, text that a spammer could not visually distinguish. For the purposes of this example, we can also assume that the PSTN number of 555-5557 is both accessible on the website and owned by the research team. Let it be further assumed that the attacker has not only harvested the above data, but will automatically feed that data into another, separate program that will make the actual calls to the supposed VoIP server. It is important to note that the SRV records that correspond to the LightningDog.com address will be configured to send all traffic to the forwarding proxy which, in turn, will send its traffic to the analysis server.

The attackers' automated calling tool would begin by initiating a connection with the VoIP server, utilizing one of the two SIP URIs that he harvested earlier. The attacker would need to initiate a connection with the forwarding proxy by sending an INVITE message to the forwarding proxy. At this point, the proxy would begin logging information on the packets that it transmits to both ends of the established stream. These logs could either be recorded on the machine that is running the proxying script, or would be sent over another connection to a different server for either more secure storage or analysis. Because the forwarding proxy is essentially a user-land script that handles every single packet that is transmitted between the attacker and the recording server, the host operating system is capable of recording, monitoring, and editing traffic as needed for the researchers.

The analysis server, configured to respond with an OK to all incoming connections would send its reply to the forwarding proxy which would then bounce that back again to the attacker, thereby completing the SIP portion of the session negotiation. After this session is established, RTP packet flow begins utilizing the already created session between the forwarding proxy and the analysis server. Throughout this entire process, the proxy is transparent to the attacker; he is led to believe that the machine he is interacting with is actually the server in question. Because of the fact that the proxy will automatically

forward any and all packets that are transmitted across SIP or RTP ports, packet scans will not reveal anything amiss should a dedicated attacker attempt to investigate the forwarding proxy.

Finally, the attacker's call data has been stored and collected upon the analysis server in both audio form as well as a collection of the packets that comprised that call. Where on a single server setup the system would simply run a list of automated scripts to derive interesting data from the recordings and then store them into a local database, on a multi-server network, the servers could collectively load-balance the influx of calls, and collectively transmit their data to another, offsite or secured database. It is important to note that the attacker's call would be routed to separate mailboxes based on the number dialed, thereby allowing the researchers to easily determine the amount of traffic, and thereby the types of techniques, that attackers are using to harvest URI data.

Recalling that the attacker had harvested the PSTN number from the bait-website as well, we can assume that the attacker would attempt to initiate a call with the analysis server. In this scenario, let us assume that the attacker is attempting to initiate this call from a VoIP system. The attacker's call would initiate at his computer travel across the IP network, until it reached the local telephone service providers ENUM servers (Rosenberg). These servers would then internally route the call using the E. 164 call numbering standard to ensure that the call reaches the correct destination, should that automated dialing script attempt to communicate through the attackers PSTN venue to the researchers VoIP connection; for such a PSTN to VoIP connection, the scenario proceeds as is detailed above. However, the caller could attempt to initiate a PSTN connection to the researcher's server using exclusively PSTN transmissions. In this case, we are able to allow the call to route directly to the server, due to the obfuscation of the internal routing network, and the service of multiple telephone service providers to allow a PSTN trunk to be purchased in a remote geographic location while routing incoming calls to yet another location. In this event, a call would directly be received and collected by the server, the overall

process transparent to the researchers, while still allowing them to modify the communication or recordings as needed. For practical intents and purposes, a call that arrives across the PSTN avenue is treated in much the same manner as a standard VoIP call.

Lastly, should the caller attempt to dial in with a request for video conferencing, that request could be accepted or denied independently of the audio component of the communication. If nothing else, the server supports the ability to reply back with a pre-made recording, thereby giving the researchers' ruse an added layer of legitimacy in the eyes of the attacker.

Related Work:

Primarily, this section discusses the literary basis for the underlying framework, along with a discussion of some of the history of related systems. Firstly, it lays out some of the original motivation, and proceeds to discuss related background information on honeynet design, ending with a quick discussion of other work that has been seen in this field.

End users are ultimately most hurt by electronic attacks, and their effects lead to a "trickle up" effect on technology; in this case, the burgeoning technology of VoIP. End users, inconvenienced already with the difficulty in learning a new technology are subjected to the harassment, fraud, and general threats those abusers of this system present. ISPs and other service providers have their systems unnecessarily taxed due to the unwanted traffic that is flowing across their networks, leading to a desire for traffic shaping and other methods to control the tremendous influx of data that spammers force service providers, from the ISP's to the hobbyist PBX provider, to endure. As a result, systems have been created to allow users and researchers who may have limited resources to capture live data from attackers as those attacks occur. Historically, the most successful of these systems have been honeynets (Krasser et al. 23-37). Comprised of interconnected networks of honeypots, honeynets are designed to capture malicious

activity and code for later detailed analysis. These networks are often closely monitored for any changes, so as to ensure that all of an attacker's activities are captured. There are two major types of honeynets, characterized by the amount of interactivity that the attacker is able to achieve with the individual machine within the network that they are attempting to compromise; divided into low and high interaction honeypots, these two classifications vary greatly in the means of their construction and the infrastructure that is required to support them.

The Low interaction honeypot is likely the most popular and the most easily created and maintained honeypot at this time. This system is a computer which hosts a suite of programs or deception services which emulate an exploitable system to a degree that it invites and deceives automated attacks from attackers. Commonly, this system is comprised of numerous minor applications which can be configured to emulate known services that are commonly exploited. These systems use a number of techniques, such as banner forging, to fool an attacker into attempting to exploit the perceived service. For example, one such program could attempt to emulate the well known and often attacked FTP service. This deceptive process would be capable of simply lying in wait for an attacker to attempt to connect to this port, and then at such a time, transmit deceptive information to fool an automated attack for the purposes of capturing its actions for later analysis. This type of honeynet is exceedingly popular due to its ease of creation and relatively low hardware requirements; however, it does suffer from a lack of robustness, and is often very quickly discovered for its true nature by a dedicated or alert attacker. Originally published alongside the honeyd suite, the field of such programs has expanded from a small collection of scripts to robust services that are capable of emulating multiple operating systems and known vulnerabilities.

Directly in contrast with the low interaction honeypot, the high interaction honeypot does away with the model of deception that the low interaction honeypot is based on, instead opting to allow attackers

to assault the hosted services that are detected during the discovery phase of an attack. This approach allows for a greater range of services to be hosted without the arduous task of creating an emulator for each piece of vulnerable software the research is centered around. Furthermore, as the attacker is interacting with precisely the service he believes he is exploiting, there is a much lower chance that an educated attacker would cease a manual attack due to the detection of a trap or other monitoring. While a low interaction honeynet will fool automated attacks and uneducated or unmotivated attackers, a competent attacker would discover the ruse fairly swiftly, and, having this knowledge, cease further attacks, thereby reducing the honeynet to be a one-shot measure; something that is highly undesirable for many research projects. In addition to these concerns, it is conceivable that an attacker could be explicitly trying to target a particular version of software that is not currently emulated, leading the attacker to either pass over the honeynet entirely or attempt to blindly exploit the honeynet with undesirable results due to failure. It is due to precisely those issues that Lightning Dog has been designed as a high-interaction honeynet.

There are a number of issues and difficulties in establishing a robust honeynet, and there are just as many theories on how to resolve these issues. Sometimes these solutions are discreet solutions to individual parts of the issues that exist, or are entire suites of tools and techniques that are dedicated to being an all-around solution. These issues revolve around the general security and anonymity of the honeynet, as should the network be discovered, or should hosts within the system be compromised, it will likely make future collected data worthless as attackers will either corrupt currently gathered data, or avoid the network altogether. Effectively, these two concerns fall into two categories: anonyminity and security (Gómez).

The achievement of keeping the system secure is somewhat different than a standard high-interaction honeynet. Whereas average high-interaction honeynets possess the inherent risk that their services

could be compromised to the degree that attackers could corrupt the recorded data or if nothing else, discover that there is a honeynet in existence at a particular address space, this system could be detected as a ruse based on the reply that is received. Should researchers attempt to imitate a foreign business, but record their response to incoming calls in English, they could be fairly easily detected by an attacker who listens to the actual replies from the victim server. Further, the design of the proxy nodes has been kept simple, in an effort to avoid having an attacker exploit the proxying system itself. Lastly, the analysis system has been hardened against standard attacks, and all hosted processes are set to run with as little permission as possible, in an effort to guard against exploits targeting the systems hosted services (Porter).

Despite the relative dearth of publications regarding this specific application of honeynet design, a few papers have been published on this specific type of honeynet application (Nassar 109-118). Most prominently, a paper created by a team of European researchers was written with the goal of creating an enterprise level solution (Nassar 109-118). This team designed an intrusion detection system for VoIP networks, and while they did not attempt to create a honeynet in the traditional sense, they laid the foundation for multiple aspects of this project. Lightning Dog differs, however, in its method of data collection, as it deliberately attempts to attract malicious traffic to its listening nodes, as opposed to attempting to act as a wall between attackers and a larger network. Furthermore, Lightning Dog takes specific means to avoid detection, going beyond the standard flow for a SIP server in that it deliberately forwards all SIP and RTP traffic between itself and the recipients, as opposed to only allowing the caller and recipient to establish a direct connection to each other. Furthermore, as opposed to other systems, this particular framework is designed to be relatively distributed, allowing a specific research group to place listing nodes without concern for geographic limitations; other systems are designed to be

deployed internally within a secured network, deliberately trying to close boarders, as opposed to open them.

Discussion and Possible Future Research

> Herein, the future direction of this research is outlaid, primarily dealing with the deployment of the system upon a larger network. Furthermore, the shortcomings of this system are described, and they primarily revolve around the difficulty of attracting traffic to the network.

To date, the research is considered a success. The system accepts calls, and testing has confirmed that it does so in an efficient and scalable manner. Further tests with SIP call and traffic generators have confirmed that the slope of load scaling remains linear up to 10,000 nodes, with the primary amount of latency the result of disk I/O and database interaction. With some time, improvements could be made in both of these areas, leading to an overall increase in the number of simulations connections that could be established. It is important to note that have been, as yet, no spam calls made to the server.

As the team has discovered, the most difficult part of the deployment of this project is not the set up or initial configuration, but instead effectively attracting traffic to the honeynet itself. Further work is being done on establishing a collection of pre-generated communities along with seeds to attract users to these communities. Until that time, this remains the primary point of weakness in this system, though that could be alleviated as telephonic spam begins to rise to meet the rate of growth in the overall industry.

At this point in the research, the team is preparing to deploy the system on the Georgia Tech campus, giving the system a chance to be used in the world of current research upon an active and large telephonic network. It is anticipated that with the larger area for which this system to monitor, attacks are more likely to be detected and captured.

As has been mentioned elsewhere in this paper, there is a relative dearth of research energy being pressed into the specific application of honeynet principal's in the world of VoIP. Despite this, however, there are other projects that aim to create IDS capacity for VoIP networks. To date, however, these other systems seem to be designed for the active detection and prevention of incoming attacks on a secured, presumably corporate network, as opposed to being a framework to invite and record, as opposed to actively defend, a network controlled by researchers. Aside from purposeful differences, these other systems to not attempt to merge and correlate the incoming attacks from both VoIP and PSTN vectors, and it is this merging of the telephonic attack avenues that ultimately make this research interesting.

Conclusions:

The ultimate goal of this research is to develop a system with which VoIP researchers and enthusiasts can quickly and effectively establish a concise format for building a cluster of machines for intercepting, recording and analyzing VoIP data that will be distributed amongst interested academic institutions. The successful implementation of a concise and efficient gathering system for relevant VoIP data will eventually lead to a more widespread effort to research the field as researchers will have a standardized method of gathering and formatting data. Finally, the success of this research will yield an open development platform for testing new software and hardware, thereby delivering an expandable platform which could organically grow and be actively developed for future use.

Seifert, Christian. "Know Your Enemy: Behind the Scenes of Malicious Web Servers." The New Zealand Honeynet Project. 7 November 2007 . The Honeynet Project. 30 Dec 2008 <http://www.honeynet.org/papers/wek>.

Nassar, M. "VoIP Honeypot Architecture." Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium 25(2005): 109-118. (Nassar 109-118)

Gómez, Diego. "Building a GenII Honeynet Gateway." Spanish Honeynet Project. 14 November, 2004. The Spanish Honeynet Project. 30 Dec 2008 <http://www.honeynor.no/docs/annual_status_report_2008.txt>.

Perez, David. "Scan of The Month 32 Write-up." Spanish Honeynet Project. Oct 27, 2004. The Spanish Honeynet Project. 30 Dec 2008 <http://www.honeynet.org/scans/scan32/>.

Doring, Christian. "Conceptual Framework for a Honeypot Solution." Honeynet Project. September 2005. German Honeynet Project. 30 Dec 2008 <http://old.honeynet.org/papers/individual/HPframework.pdf>.

Holz, Thorsten. "Detecting Hone." Detecting Honeypots and Other Syspicious Enviroments. The French and German Honeynet Project. 30 Dec 2008 <http://old.honeynet.org/papers/individual/DefeatingHPs-IAW05.pdf>.

Krasser, Sven, Julian Grizzard , Henry Owen , and John Levine . "The use of honeynets to increase computer network security and user awareness ." Journal of Security Education . 1(2005): 23-37.

Honeynet Learning Applying problem and case-based approach in IT security education through the use of honeynets. It's scheduled for publication in the ACM InRoads journal in June 2006. [Phillipine Honeynet Project].

Rosenberg J. et al. SIP: Session Initiation Protocol, RFC 3261, June 2002

Niccolini, Saverio. "Holistic VoIP Intrusion Detection & Prevention System." *IPTComm 2007 Telecommunications in the Internet Age* (2007): 1-9. Print.

Fiedler, Jens et. al." VoIP Defender: Highly Scalable SIP-based Security Architecture" *IPTComm 2007 Telecommunications in the Internet Age* (2007) pages 11-17. Print

VoIPSA, "Voip security and privacy threat taxonomy," Public Realease 1.0, Oct 2005, http:Iwww.voipsa.org/Activities /OIPSA Threat Taxonomy 0.1 .pdf.

Abdelnur, H, V. Cridlig, J. Bourdellon, and R. State. "Voip security management." *18th meeting of the Network Management Research Group*. (2005): Print.

Porter, T. *PracticalVoIPSecurity*. Rockland, MA: Syngress Publishing, 2006. Print.

Arnold, Jon. "VoIP on the verge." *Telecommuncations Online* 01 Nov 2004 Web.27 Apr 2009. <http://www.telecoms-mag.com/Americas/article.asp?HH_ID=AR_539>.

Callahan, Renee . "Businesses Move To Voice-Over-IP." *Forbes.com*. 03 Apr 2009. Forbes.com. 27 Apr 2009 <http://www.forbes.com/2008/12/09/skype-vonage-ringcentral_leadership_clayton_in_rc_1209claytonchristensen_inl.html>.

"VoIP Review". VoIP Review, LLC. http://www.voipreview.org. Retrieved on 2008-12-06.

"RFC 3969 - The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)". The Internet Society. December 1, 2004. http://www.packetizer.com/rfc/rfc3969/. Retrieved on 2009-01-21.

"Application-level Network Interoperability and the Evolution of IMS". TMCnet.com. May 24, 2006. http://ipcommunications.tmcnet.com/hot-topics/MCP/articles/1311-application-level-network-interoperability-the-evolution-ims.htm. Retrieved on 2009-01-21.

Packetcable Implementation P557 - Jeff Riddel - ISBN 1587051818